In the Specification:

Please add the following new paragraphs after paragraph 27 under the Brief Description of the Drawings section of the application:

Figure 4 is a functional diagram illustrating the address generation for a first input to a butterfly according to an embodiment of the present invention.

Figure 5 is a functional diagram illustrating the address generation for the second input to the same butterfly as discussed with reference to Figure 4 according to an embodiment of the present invention.

Figure 6 is a functional block diagram illustrating a multiprocessor system according to one embodiment of the present invention.

Please add the following paragraphs after paragraph 32 in the Detailed Description section of the present application:

The desired assignment of operand addresses may be achieved by deriving the address of the first operand in the operand pair of the butterfly corresponding to the "ith" stage of the computation from the address of the corresponding operand in the previous stage by inserting a "0" in the "(i+1)th" bit position of the address. The address of the second operand is derived by inserting a "1" in the "(i+1)th" bit position of the operand address. The computing of twiddle factors for the butterfly computations at each processor may be done by initializing a counter and then incrementing the counter by a value corresponding to the number of processors "P" and appending the result with a specified number of "0"s. This describes the algorithm for generating the input addresses for the three inputs required for computing butterfly operations, namely the addresses for data inputs and one twiddle factor address.

This address generation is illustrated diagrammatically in Figures 4 and 5. Let the size of the FFT/IFFT be N, the number of stages = Log2N = K, and let the current stage number be i, where i = 0,1,2...(K-1). Now consider a sequential counter of (K-1) bits as illustrated in Figures 4 and 5. At every stage, this counter counts up to (N/2-1), starting from zero. For generating addresses of inputs to the butterfly in a stage i, the

address for the first input # 1 is generated by introducing a '0' in the (i+1)th position from the LSB of the counter as shown in Figure 4. Similarly, the address for the second input # 2 is generated by introducing a '1' in the (i+1)th position from LSB of the counter as shown in Figure 5.

For generating the address of the twiddle factor in a stage 'i' a separate counter is used with the number of bits equal to (i+2) on each processor j, where j = 0,1,2...(P-1) with P being the number of processors in the system. In each processor j, the counter is initialized with the value j and {(K-1) – (i+2)} zeroes are appended to the counter value to get the twiddle factor address. The counter is then incremented by P and appended with {(K-1) – (i+2)} zeroes to get the twiddle factor address of the next butterfly in stage.

Please amend paragraph 33 of the Detailed Description section of the present application as follows:

[33] FIG. 3 shows a 4-processor implementation for the 16-point FFT ~~using this invention~~according to an embodiment of the present invention. Different line colors or characteristics represent computations in each of the 4 processors. Figure 6 is a functional block diagram illustrating a multiprocessor system 600 for implementing the 16-point FFT of Figure 3 according to one embodiment of the present invention. Input data DATA IN to be transformed is input to a memory system 602 including four memories 604a-d, each memory storing data for a corresponding processor 606a-d. An address generation circuitry 608 address circuitry distributes the computation of the butterfly computational blocks in all stages subsequent to the first log2P states among the plurality of processors 606 such that each chain of cascaded butterfly computational blocks in the transform are coupled in series and are computed by the same processor. The address generation circuitry 608 derives the address of the first operand in an operand pair corresponding to the "ith" stage of the computation from the address of the corresponding operand in the previous stage by inserting a "0" in the "(i+1)th" bit position of the address, and derives the address of the second operand by inserting a "1" in the "(i+1)th" bit position of the operand address. The address generation circuitry

608 computes twiddle factors for the butterfly computations in each processor P by initializing a counter 610 and then incrementing this counter by a value corresponding to the number of processors 606 and appending the result with a specified number of "0"s.

Please add the following paragraphs after paragraph 38 of the Detailed Description section of the application:

According to one embodiment of the present invention, a scalable method for implementing FFT/IFFT computations in multiprocessor architectures provides improved throughput by eliminating the need for inter-processor communication after the computation of the first "$\log_2 P$" stages of the FFT/IFFT computations for a multiprocessor architecture including an implementation using "P" processing elements. The method includes computing each butterfly of the first "$\log_2 P$" stages on either a single processing element or on each of the "P" processing elements simultaneously and distributing the computation of the butterflies in all the subsequent stages among the "P" processors such that each chain of cascaded butterflies consisting of those butterflies that have inputs and outputs connected together, are processed by the same processor.

In one embodiment of this method the distributing of the computation of the butterflies subsequent to the first "$\log_2 P$" butterflies is achieved by assigning operand addresses of each set of butterfly operands to each processor in such a manner that the butterfly is processed by the same processor that computed the connected butterfly of the previous stage in the same chain of butterflies. The desired assignment of operand addresses may be achieved by deriving the address of the first operand in the operand pair corresponding to the "i^{th}" stage of the computation from the address of the corresponding operand in the previous stage by inserting a "0" in the "$(i+1)^{th}$" bit position of the address, while the address of the second operand is derived by inserting a "1" in the "$(i+1)^{th}$" bit position of the operand address. This embodiment may further include the computing of twiddle factors for the butterfly computations at each processor by

initializing a counter and then incrementing it by a value corresponding to the number of processors "P" and appending the result with a specified number of "0"s.

In another embodiment of the present invention, a system for obtaining scalable implementation of FFT/IFFT computations in multiprocessor architectures provides improved throughput by eliminating the need for inter-processor communication after the computation of the first "$\log_2 P$" stages for an implementation using "P" processing elements. The system includes a means for computing each butterfly of the first "$\log_2 P$" stages on either a single processor or each of the "P" processors simultaneously and an addressing means for distributing the computation of the butterflies in all the subsequent stages among the "P" processors such that each chain of cascaded butterflies consisting of those butterflies that have inputs and outputs connected together, are processed by the same processor.

In one embodiment the addressing means includes addresses generation means for deriving the operand addresses of the butterflies subsequent to the first "log2P" butterflies in such a manner that the butterfly is processed by the same processor that computed the connected butterfly of the previous stage in the same chain of butterflies. The address generation means may be a computing mechanism for deriving the address of the first operand in the operand pair corresponding to the "ith" stage of the computation from the address of the corresponding operand in the previous stage by inserting a "0" in the "(i+1)th" bit position of the address, and deriving the address of the second operand by inserting a "1" in the "(i+1)th" bit position of the operand address. The system may further include a computing mechanism for address generation of twiddle factors for each butterfly on the corresponding processor.

In one embodiment, a method of performing a fast Fourier transform or inverse fast Fourier transform on a plurality of inputs to generate a plurality of outputs is performed on a plurality of processors and each transform includes a plurality of stages containing at least one butterfly computational block. This embodiment may include calculating the butterfly computational blocks for the first log2(P) stages of the transform on a single one of the processors or on a plurality of the processors operating in parallel and calculating chains of butterfly computational blocks corresponding to the

subsequent stages of the transform within each of the processors, each chain of butterfly computational blocks that is calculated in a respective processor having inputs and outputs coupled in series.

The first log2(P) stages of the transform may be calculated on all of the processors operating in parallel. This embodiment may be implemented on two processors, with the first two stages of a radix-2 fast Fourier transform or inverse fast Fourier transform calculated as a single radix-4 stage, and the subsequent stages of the transform are computed as radix-2 stages. The chains may comprise a single loop that iterates N/2 *(log2(N/2))/(number of processors) times. Each butterfly computational block may include a plurality of operands each having an associated address. Calculating chains of butterfly computational blocks corresponding to the subsequent stages may include assigning addresses to each of the operands so that each butterfly block in a chain is calculated in the same processor. Each butterfly computational block may include a pair of operands and the operand addresses of these operands may be assigned by deriving the address of the first operand in the operand pair corresponding to the "ith" stage of the calculation in the chain from the address of the corresponding operand in the previous stage by inserting a "0" in the "(i+1)th" bit position of the operand address, and deriving the operand address of the second operand by inserting a "1" in the "(i+1)th" bit position of the operand address. This embodiment may further include initializing a counter and then incrementing the counter by a value corresponding to the number of processors and appending the result with a specified number of "0"s to compute the twiddle factors for each butterfly computational block.

According to another embodiment of the present invention, a processor system includes a plurality of processors operable to execute a fast Fourier transform or inverse fast Fourier transform algorithm on a plurality of inputs to generate a plurality of outputs, each transform including a plurality of stages containing at least one butterfly computational block, and the processors operable to the butterfly computational block for the first "log2P" stages of the transform on either a single one of the processors or on a plurality of the processors operating in parallel. Address circuitry is operable to distribute the computation of the butterfly computational blocks in all stages subsequent to the first log2P states among the plurality of processors such that each chain of

cascaded butterfly computational blocks in the transform are coupled in series and are computed by the same processor.

The address circuitry may be further operable to derive operand addresses for each of the butterfly blocks subsequent to the first "log2P" butterfly blocks so that each of the butterfly computational blocks is computed by the same processor that computed a butterfly computational block of the previous stage in the same chain of butterfly computational blocks. Each butterfly computational block may include a pair of operands and the address circuitry may assign operand addresses of these operands by deriving the address of the first operand in the operand pair corresponding to the "ith" stage of the calculation in the chain from the address of the corresponding operand in the previous stage by inserting a "0" in the "(i+1)th" bit position of the operand address, and deriving the operand address of the second operand by inserting a "1" in the "(i+1)th" bit position of the operand address. The processors may further comprise a counter that is initialized and then incremented by a value corresponding to the number of processors, an output of the counter being appended with a specified number of "0"s to compute twiddle factors for each butterfly computational block. Each of the processors may include a digital signal processor. An electronic system may include this processor system where the electronic system is a communications system. Each of the processors may be a digital signal processor.